Big Data in Finance

Lecture 2: Linear Regression Methods

Dr Daniele Bianchi

Spring 2026

Contents

Overview

In the first lecture, we established the fundamental framework for machine learning: we seek to learn a prediction function $\widehat{f}$ from data, and we face a tradeoff between bias and variance. We emphasized that models must be evaluated out-of-sample and that financial applications require special care due to the time-series nature of the data.

This lecture addresses a question that arises immediately in practice: what happens when we have many potential predictors? In finance, this situation is ubiquitous. A portfolio manager might have access to dozens or even hundreds of firm characteristics—size, valuation ratios, momentum signals, profitability measures, and more—all of which might plausibly predict returns. A credit analyst might have hundreds of borrower attributes to consider. How do we build a model when the number of features is large relative to the number of observations?

The answer, as we will see, involves deliberately introducing bias into our estimates. This sounds counterintuitive — why would we want biased estimates? — but the bias-variance trade-off shows that accepting some bias can dramatically reduce variance, leading to better overall predictions. The methods we study today — Ridge regression, Lasso, and Elastic Net — all work by "shrinking" coefficient estimates toward zero, thereby trading off a bit of bias for a substantial reduction in variance.

Sections 1 to 4 develop the theory: why does ordinary least squares (OLS) fail with many predictors, and how do penalized regression methods address this failure? Sections 5 to 7 turn to applications: how do we select the tuning parameters in practice, and what happens when we apply these methods to real financial prediction problems?

1 The Challenge of High-Dimensional Prediction

Let us begin by recalling the linear regression framework. We observe n data points, each consisting of an outcome y_i and a vector of p predictors $x_i = (x_{i1}, \ldots, x_{ip})$. The linear model assumes:

$$y_i = x_i'\beta + \epsilon_i, \qquad i = 1, \ldots, n$$

where $\beta = (\beta_1, \ldots, \beta_p)'$ is the vector of coefficients we want to estimate, and ϵ_i is the error term with $\mathbb{E}[\epsilon_i] = 0$. In matrix notation, we write $y = X\beta + \epsilon$, where y is $n \times 1$, X is $n \times p$, and ϵ is $n \times 1$.

1.1 OLS: A Quick Review

The ordinary least squares estimator minimizes the sum of squared residuals:

$$\widehat{\beta}^{OLS} = \arg\min_{\beta} \sum_{i=1}^{n} (y_i - x_i'\beta)^2 = \arg\min_{\beta} \|y - X\beta\|^2$$

This has the familiar closed-form solution:

$$\widehat{\beta}^{OLS} = (X'X)^{-1}X'y$$

Under standard assumptions—in particular, that $\mathbb{E}[\epsilon|X] = 0$ and the errors have constant variance—OLS has beautiful properties. It is unbiased: $\mathbb{E}[\widehat{\beta}^{OLS}] = \beta$. And by the Gauss-Markov theorem, it is the best linear unbiased estimator (BLUE), meaning no other linear unbiased estimator has lower variance.

These properties make OLS the workhorse of econometrics when the number of predictors is small relative to the sample size. But they tell us nothing about what happens when p grows large.

1.2 When OLS Fails

Consider a concrete example. Suppose you want to predict stock returns using firm characteristics, and you have access to 100 characteristics: various size measures, valuation ratios, momentum signals, profitability metrics, and so on. If your sample consists of 60 months of data, you have $p = 100$ predictors and $n = 60$ observations.

This is a disaster for OLS. With more predictors than observations ($p > n$), the matrix $X'X$ is not invertible, and OLS simply cannot compute a unique solution. The system of equations is "underdetermined", meaning that there are infinitely many coefficient vectors that fit the training data perfectly.

But the problem starts well before p exceeds n. Even when p is merely "large relative to n"—say, $p = 50$ with $n = 100$—OLS performs poorly. Why?

The key insight comes from the variance of the OLS estimator:

$$\text{Var}(\widehat{\beta}^{OLS}) = \sigma^2(X'X)^{-1}$$

As p grows and approaches n, the matrix $(X'X)$ becomes increasingly ill-conditioned—its smallest eigenvalues approach zero, and inverting it amplifies noise in the data. The result is that coefficient estimates become wildly unstable. Small changes in the data lead to large swings in $\widehat{\beta}$. The model fits the training data well—perhaps too well—but generalizes poorly to new data.

This is the overfitting problem in its starkest form. With many predictors, OLS finds combinations of predictors that happen to fit the particular noise realizations in the training sample, but these spurious patterns do not persist out of sample.

1.3 The Intuition for Shrinkage

How can we address this problem? The key insight comes from the bias-variance decomposition we studied in Lecture 1:

$$\text{MSE}(\widehat{\beta}) = \text{Bias}(\widehat{\beta})^2 + \text{Var}(\widehat{\beta})$$

OLS achieves zero bias but can have enormous variance when p is large. What if we accepted some bias in exchange for a substantial reduction in variance? If the variance reduction is sufficiently large, the total mean-squared error may decrease.

This is the idea behind **regularization** or **shrinkage**. Instead of letting coefficients take whatever values best fit the training data, we constrain them—we "shrink" them toward zero. Coefficients can only be large if strongly supported by the data. Weak or spurious signals get attenuated.

The intuition is one of **disciplined skepticism**. Without regularization, OLS is prone to finding any pattern in the data, no matter how implausible. With regularization, we impose a prior belief that most coefficients should be small. The data can override this prior, but only with sufficient evidence.

In finance, this skepticism is particularly appropriate. We know that markets are competitive and that obvious predictive patterns tend to be arbitraged away. Finding a coefficient of 0.5 in a sample of 60 months should not make us confident that the true coefficient is 0.5 — it could easily be noise. Shrinkage formalizes this healthy skepticism.

2 Ridge Regression

The first regularization method we study is **Ridge regression**, which adds a penalty on the sum of squared coefficients to the OLS objective.

2.1 The Ridge Objective

Ridge regression solves:

$$\widehat{\beta}^{Ridge} = \arg\min_{\beta} \left\{ \sum_{i=1}^{n} (y_i - x_i'\beta)^2 + \lambda \sum_{j=1}^{p} \beta_j^2 \right\}$$

The first term is the familiar residual sum of squares (RSS)—we still want to fit the data. The second term is the **Ridge penalty**: the sum of squared coefficients, multiplied by a tuning parameter $\lambda \geq 0$.

The penalty term $\sum_{j=1}^{p} \beta_j^2 = \|\beta\|_2^2$ is the squared L_2 norm of the coefficient vector. It penalizes large coefficients: if β_j is large, it contributes substantially to the penalty, and the optimization will try to shrink it.

The tuning parameter λ controls the strength of regularization:

- When $\lambda = 0$, there is no penalty, and we recover OLS.
- As $\lambda \to \infty$, the penalty dominates, and all coefficients are shrunk toward zero.
- Intermediate values of λ balance fit (low RSS) against complexity (small coefficients).

A key advantage of Ridge regression is that it has a closed-form solution:

$$\widehat{\beta}^{Ridge} = (X'X + \lambda I)^{-1} X'y$$

Compare this to OLS: $\widehat{\beta}^{OLS} = (X'X)^{-1}X'y$. The only difference is that we add λI to $X'X$ before inverting.

This small modification has profound consequences. Adding λI to $X'X$ ensures the matrix is always invertible, even when $p > n$ or when predictors are highly correlated. The smallest eigenvalue of $X'X + \lambda I$ is at least λ, so the matrix is well-conditioned. This "stabilizes" the estimation and prevents the wild coefficient swings that plague OLS with many predictors.

2.2 Geometric Interpretation

It is helpful to think about Ridge regression geometrically. The optimization problem can be written equivalently as a constrained problem:

$$\min_{\beta} \sum_{i=1}^{n} (y_i - x_i'\beta)^2 \qquad \text{subject to} \qquad \sum_{j=1}^{p} \beta_j^2 \leq t$$

The constraint $\sum_j \beta_j^2 \leq t$ defines a sphere in p-dimensional space (a circle in two dimensions). We are minimizing RSS subject to the coefficients lying within this sphere.

Imagine the contours of constant RSS as ellipses centered at the OLS solution $\widehat{\beta}^{OLS}$. The Ridge solution is where the smallest RSS contour touches the constraint sphere. Unless $\widehat{\beta}^{OLS}$ happens to lie within the sphere (meaning the constraint is not binding), the Ridge solution will be strictly inside the sphere—closer to zero than OLS.

2.3 What Ridge Does: Proportional Shrinkage

Ridge regression shrinks all coefficients toward zero, but it does so proportionally—coefficients that are large under OLS remain relatively large under Ridge, just scaled down. Importantly,

Ridge never sets a coefficient exactly to zero. All p predictors remain in the model; they just have smaller coefficients.

This proportional shrinkage has a specific interpretation. When predictors are correlated, OLS tends to give one predictor a large positive coefficient and another a large negative coefficient, with the effects partially canceling. Ridge spreads the weight more evenly among correlated predictors, leading to more stable estimates.

2.4 The Bias-Variance Tradeoff for Ridge

Ridge regression is biased: $\mathbb{E}[\widehat{\beta}^{Ridge}] \neq \beta$. The coefficients are systematically shrunk toward zero, so on average, we underestimate their magnitude.

But Ridge has lower variance than OLS. Shrinkage stabilizes the estimates, making them less sensitive to the specific training sample.

The tradeoff depends on λ:

- Small λ: Low bias, high variance (close to OLS)
- Large λ: High bias, low variance (coefficients close to zero)
- Optimal λ: Minimizes total MSE

A plot of MSE against λ typically shows bias^2 increasing with λ (more shrinkage means more bias) and variance decreasing with λ (more shrinkage means more stability). The total MSE is U-shaped, with a minimum at some intermediate λ^*.

2.5 Practical Consideration: Feature Scaling

There is one important practical consideration with Ridge regression: the penalty treats all coefficients equally, but coefficients depend on the scale of the corresponding predictor. If one predictor is measured in dollars and another in millions of dollars, their coefficients will have very different magnitudes, and the penalty will affect them unequally.

The solution is to **standardize** all predictors before estimation:

$$\tilde{x}_{ij} = \frac{x_{ij} - \bar{x}_j}{s_j}$$

where $\bar{x}_j$ is the sample mean and s_j is the sample standard deviation of predictor j. After standardization, all predictors have mean zero and variance one, so their coefficients are on comparable scales. In practice, this is handled automatically by software.

3 Lasso Regression

Ridge regression mitigates the variance problem, but it has a limitation: it keeps all predictors in the model. If you start with 100 characteristics, your Ridge model still uses all 100, just with smaller coefficients. Sometimes we want something more: a sparse model that uses only the most important predictors.

3.1 The Lasso Objective

The **Lasso** (Least Absolute Shrinkage and Selection Operator), introduced by Tibshirani (1996), achieves sparsity by changing the penalty from squared coefficients to absolute values:

$$\widehat{\beta}^{Lasso} = \arg\min_{\beta} \left\{ \sum_{i=1}^{n} (y_i - x_i'\beta)^2 + \lambda \sum_{j=1}^{p} |\beta_j| \right\}$$

The penalty $\sum_{j=1}^{p} |\beta_j| = \|\beta\|_1$ is the L_1 norm of the coefficient vector. This seemingly small change—from β_j^2 to $|\beta_j|$—has dramatic consequences.

The crucial property of Lasso is that it can set coefficients exactly to zero. With a sufficiently large λ, many coefficients will be exactly zero, not just small. This means Lasso performs automatic **variable selection**: it chooses which predictors to include and which to exclude.

An intuitive way to understand this is through the "soft thresholding" rule. For each coefficient, Lasso essentially asks: "Is the signal strong enough to survive the penalty?" If the OLS coefficient is small (in absolute value less than λ), Lasso sets it to zero. If the OLS coefficient is large enough, Lasso keeps it but shrinks it toward zero by λ. Weak signals are eliminated; strong signals survive but are attenuated.

3.2 Lasso Properties

The sparsity of Lasso has both advantages and limitations.

Advantages:

- Automatic variable selection: We get an interpretable model that identifies "important" predictors.
- Can handle $p > n$: Lasso selects at most $\min(n, p)$ non-zero coefficients.
- Sparse models may generalize better when the true relationship is sparse.

Limitations:

- With correlated predictors, Lasso tends to select one arbitrarily and set the others to zero. This can be unstable—different samples might select different predictors from a correlated group.
- The selected model can change substantially with small changes in the data.
- Lasso selects at most n variables, which is a limitation when $p \gg n$ and many variables are relevant.

Unlike Ridge, Lasso does not have a closed-form solution. The L_1 penalty $|\beta_j|$ is not differentiable at zero, so we cannot simply set a derivative to zero and solve. Instead, Lasso requires iterative optimization algorithms, typically coordinate descent, which updates one coefficient at a time. Modern implementations are very efficient, so this is not a practical limitation.

4 Elastic Net

Ridge and Lasso each have strengths: Ridge handles correlated predictors well and produces stable estimates; Lasso performs variable selection and yields sparse models. Can we get the best of both worlds?

4.1 The Elastic Net Objective

The **Elastic Net**, introduced by Zou and Hastie (2005), combines both penalties:

$$\widehat{\beta}^{EN} = \arg\min_{\beta} \left\{ \sum_{i=1}^{n} (y_i - x_i'\beta)^2 + \lambda \left[\alpha \sum_{j=1}^{p} |\beta_j| + (1-\alpha) \sum_{j=1}^{p} \beta_j^2 \right] \right\}$$

There are now two hyperparameters:

- $\lambda \geq 0$: The overall penalty strength, as before.
- $\alpha \in [0, 1]$: The mixing parameter between L_1 and L_2 penalties.

When $\alpha = 1$, Elastic Net reduces to Lasso. When $\alpha = 0$, it reduces to Ridge. Intermediate values give a combination.

4.2 Elastic Net Properties

Elastic Net inherits advantages from both methods:

From Lasso (the L_1 component):

- Sparse solutions—can set coefficients exactly to zero.
- Automatic variable selection.

From Ridge (the L_2 component):

- Grouping effect: Correlated predictors tend to be selected or excluded together, rather than one being arbitrarily chosen.
- More stable than pure Lasso.
- Can select more than n variables (unlike Lasso).

In practice, Elastic Net is often the best default choice for high-dimensional problems, particularly when predictors are correlated—as they typically are in finance.

The following table summarizes the key properties of the methods we have studied:

Method	**Penalty**	**Sparse?**	**Closed-form?**	**Handles Correlation?**
OLS	None	No	Yes	Poorly
Ridge	L_2	No	Yes	Well
Lasso	L_1	Yes	No	Poorly
Elastic Net	$L_1 + L_2$	Yes	No	Well

Practical guidance:

- Few predictors, low correlation: OLS may suffice.
- Many correlated predictors, no need for sparsity: Ridge.
- Want variable selection, low correlation: Lasso.
- High-dimensional, correlated predictors: Elastic Net.

5 Selecting the Tuning Parameter

All the methods we have discussed depend on the tuning parameter λ (and, for Elastic Net, also α). How do we choose these values?

5.1 The Problem with Training Error

One might think: just try different values of λ and pick the one with the lowest prediction error. But which prediction error? If we use the training error — the error on the same data used to fit the model — we will always prefer $\lambda = 0$, which yields OLS. Training error always decreases (or remains constant) as we reduce regularization.

This is the problem we anticipated in Lecture 1: training error is not a reliable guide to out-of-sample performance. We need a proxy for how the model will perform on new data.

5.2 Time-Series Cross-Validation

The standard approach is cross-validation, but for financial data we must preserve the time ordering. Standard K-fold CV randomly assigns observations to folds, which would create look-ahead bias: we might train on 2020 data and "validate" on 2018 data, using future information to predict the past.

Time-series cross-validation maintains temporal ordering. The key principle: always train on past data, test on future data.

In an **expanding window** approach:

1. Start with an initial training period (e.g., the first 10 years).
2. Fit the model and predict the next period.
3. Add that period to the training data and repeat.

In a **rolling window** approach:

1. Use a fixed-size window (e.g., 10 years) for training.
2. Predict the next period.
3. Slide the window forward, dropping old data and adding new data.

The expanding window uses all available past data, which is appropriate if relationships are stable over time. The rolling window discards old data, which helps if relationships change (regime shifts).

5.3 The Cross-Validation Procedure

To select λ via time-series CV:

1. **Define a grid of candidate values.** Typically, λ values are spaced on a logarithmic scale: $\{0.001, 0.01, 0.1, 1, 10, 100\}$.
2. **For each λ, compute the CV error.** Split the training data into K folds (respecting time order). For each fold, train on earlier folds, predict on the current fold, and record the MSE. Average across folds.
3. **Compute standard errors.** The K folds give K different error estimates. The standard error measures how much these vary.
4. **Select the optimal λ.** Two common approaches:
   - $\lambda_{\min}$: The value that minimizes mean CV error.
   - λ_{1SE}: The largest λ whose CV error is within one standard error of the minimum.

5.4 The One-Standard-Error Rule

The **one-standard-error rule** is a conservative heuristic that protects against overfitting to the validation set. The idea is simple: if two models have CV errors that are statistically indistinguishable (within one standard error), prefer the simpler model (larger λ, more regularization).

Why? The model that minimizes CV error on this particular data might be overfitting to noise in the validation folds. A slightly simpler model, while having marginally higher CV error, may generalize better to truly new data.

In practice, try both $\lambda_{\min}$ and λ_{1SE} and compare on a final holdout set. Often they perform similarly, and the more regularized model has other advantages: simpler interpretation, more stable predictions, lower turnover in a trading context.

6 Timing the Equity Market

We now turn to our first application: can we predict the aggregate stock market? Market timing is the holy grail of active investing. If you could predict when stocks will outperform bonds, you could shift your allocation accordingly and earn superior returns. The economic stakes are enormous.

The academic literature has proposed many predictors: valuation ratios (dividend-price, earnings-price, book-to-market), interest rate variables (Treasury bill rate, term spread, default spread), and others (stock variance, inflation, net equity expansion). Each has some theoretical motivation and has shown in-sample predictive power in various studies.

The prediction model is a time-series regression:

$$r_{m,t+1} = \alpha + x_t'\beta + \epsilon_{t+1}$$

where $r_{m,t+1}$ is the market excess return in month $t+1$, and x_t is a vector of predictors known at time t.

6.1 The Sobering Evidence

Goyal and Welch (2008) conducted a comprehensive study of equity premium prediction. They tested all major predictors from the literature using proper out-of-sample evaluation (expanding-window estimation). Their sobering conclusion: most predictors fail out-of-sample. Variables that show significant in-sample predictive power often have negative out-of-sample R^2—meaning they perform *worse* than simply predicting the historical mean.

The benchmark is deceptively simple:

$$\widehat{r}_{t+1} = \bar{r}_t = \frac{1}{t}\sum_{s=1}^{t} r_s$$

This "historical mean" forecast ignores all predictor variables; it simply predicts that next month's return will equal the average of all past returns. Yet this naive benchmark proves surprisingly hard to beat.

Goyal, Welch, and Zafirov (2024) updated this analysis with 15 additional years of data. Their conclusions largely confirm the original findings: market timing remains very difficult.

6.2 Out-of-Sample R^2

The key metric for evaluating predictive accuracy is the out-of-sample R^2:

$$R^2_{OS} = 1 - \frac{\sum_{t=t_0}^{T-1}(r_{t+1} - \widehat{r}_{t+1})^2}{\sum_{t=t_0}^{T-1}(r_{t+1} - \bar{r}_t)^2}$$

This compares the model's prediction errors to the historical mean's prediction errors. If $R^2_{OS} > 0$, the model beats the mean; if $R^2_{OS} < 0$, the model is worse than the mean.

Note that unlike in-sample R^2, which is always non-negative, out-of-sample R^2 can be negative. A negative R^2_{OS} means you would have been better off ignoring all your sophisticated predictors and just predicting the simple average.

Given the evidence that OLS fails out-of-sample, can regularized methods do better? We examine this using the Goyal, Welch, and Zafirov (2024) dataset: monthly S&P 500 excess returns from 1956 to 2021, with 25 predictor variables.

The results are illuminating:

Model	R^2_{OS} (%)
OLS	-10.69
Ridge	-1.38
Lasso	-0.11
Elastic Net	-0.17

All models have negative R^2_{OS}—none beats the historical mean. But the magnitudes differ dramatically. OLS is catastrophically bad, with $R^2_{OS} = -10.69\%$. It overfits so severely that its predictions make things much worse than ignoring the predictors entirely.

Regularized methods are far better. Lasso achieves $R^2_{OS} = -0.11\%$, essentially matching the historical mean. Ridge is slightly worse at -1.38% but still vastly better than OLS.

6.3 Why OLS Fails So Badly

The OLS disaster is a textbook example of overfitting. With 25 predictors and limited data, OLS finds patterns that fit the historical data but do not persist in the future. When these spurious patterns fail out-of-sample, the model makes wild predictions that are worse than useless.

The regularized methods constrain the coefficients, preventing the model from chasing noise. Lasso, in particular, often sets most coefficients to zero, effectively reducing them to near the historical mean. This might seem disappointing — we are not finding much predictability — but it is honest: if predictability is weak, we should not pretend otherwise.

6.4 From Predictions to Portfolios

Statistical accuracy (R^2_{OS}) is not the only metric that matters. We ultimately care about economic value: can these predictions improve portfolio performance?

A standard market-timing strategy uses forecasts to allocate between stocks and a risk-free asset. The mean-variance optimal weight is:

$$\omega_t = \frac{1}{\gamma} \frac{\widehat{r}_{t+1}}{\widehat{\sigma}^2_t}$$

where γ is risk aversion (typically 5), $\widehat{r}_{t+1}$ is the predicted excess return, and $\widehat{\sigma}^2_t$ is estimated variance. Positive predictions lead to equity allocation; negative predictions lead to cash or short positions.

The performance results are striking:

Strategy	**Ann. Return (%)**	**Sharpe Ratio**	**Turnover**
Buy & Hold	6.86	0.46	0.00
OLS	10.06	0.60	7.77
Ridge	11.14	0.70	6.34
Lasso	4.14	0.38	0.28

Before transaction costs, both OLS and Ridge appear to beat buy-and-hold. But OLS has enormous turnover (7.77 times per year), meaning it trades aggressively based on its (overfit) predictions. Ridge has somewhat lower turnover (6.34), and Lasso barely trades at all (0.28).

When we account for realistic transaction costs (50 basis points per trade), the picture changes:

Strategy	Ann. Return Net (%)	Sharpe Net
Buy & Hold	6.86	0.46
OLS	6.17	0.37
Ridge	7.97	0.50

OLS now underperforms buy-and-hold after costs. Its aggressive trading destroys the gross returns. Ridge, with more stable predictions and lower turnover, still beats buy-and-hold net of costs.

Several lessons emerge from this analysis:

Regularization is essential. OLS catastrophically overfits with many predictors. Even though $p = 25$ is much less than $n \approx 700$, the signal is so weak that OLS finds spurious patterns.

Statistical and economic performance can diverge. Lasso has the best R^2_{OS} (closest to zero) but the worst portfolio performance. Its aggressive shrinkage essentially reduces to the historical mean, which is statistically safe but economically uninteresting. Ridge has a lower R^2_{OS} but higher portfolio returns — its predictions are riskier.

Transaction costs matter enormously. The most aggressive model (OLS) looks good before costs but bad after. More stable predictions (Ridge) preserve value better.

Market timing is genuinely hard. Even with regularization and proper validation, we struggle to beat a simple buy-and-hold strategy. This is consistent with market efficiency: if timing were easy, everyone would do it, and the opportunity would disappear.

7 Cross-Sectional Return Prediction

Our second application shifts from timing the market to selecting stocks. Can we predict which stocks will outperform?

Cross-sectional return prediction uses firm characteristics to forecast individual stock returns. The model is:

$$r_{i,t+1} = x'_{i,t}\beta + \epsilon_{i,t+1}$$

where $r_{i,t+1}$ is stock i's return in month $t+1$, and $x_{i,t}$ is a vector of characteristics (size, value, momentum, etc.) observed at time t.

This setting differs from market timing in important ways:

- Many more observations: With thousands of stocks and decades of data, we have millions of stock-month observations.
- Different goal: Instead of timing the overall market, we want to rank stocks and tilt portfolios toward predicted outperformers.

7.1 The Kozak, Nagel, and Santosh (2020) Approach

Kozak, Nagel, and Santosh (2020) study cross-sectional prediction with many characteristics. They argue that regularization—particularly Ridge—embodies economically sensible skepticism about the predictive power of individual characteristics.

Their key insight: the academic literature has documented over 100 characteristics that purportedly predict returns. Many of these are likely spurious (data mining). Rather than trying to identify the "true" sparse set of predictors (which may not exist), we should use all characteristics but shrink their coefficients, letting the data determine how much weight to give each one.

7.2 Implementation and Results

The standard workflow:

1. At each time t, estimate the model using historical data.
2. Generate predictions $\widehat{r}_{i,t+1}$ for all stocks.
3. Rank stocks by predicted return.
4. Form a long-short portfolio: buy the top decile, sell the bottom decile.
5. Observe realized returns and move to $t+1$.

Hyperparameters (λ) are selected via time-series cross-validation on the training data, exactly as we discussed earlier.

The findings are more encouraging than for market timing:

- Ridge substantially outperforms OLS out-of-sample.
- Optimal λ implies significant shrinkage—the data supports skepticism about most characteristics.
- Long-short portfolios achieve Sharpe ratios above 1.
- Results are robust across different sample periods.

The contrast with market timing is instructive. Cross-sectional prediction benefits from vastly more data (many stocks, not just one market index) and exploits differences across stocks rather than trying to predict the overall level. The signal-to-noise ratio, while still low, appears higher than in market timing.

7.3 Practical Considerations

Several practical issues arise in implementation:

Transaction costs. ML strategies can have high turnover if predictions change substantially each month. Trading costs erode returns. Some practitioners constrain turnover directly or use predictions that change more slowly.

Capacity constraints. ML-based strategies often work best in smaller stocks, which have more mispricing but also less liquidity. Large institutional investors may find limited capacity.

Data quality. Point-in-time data is essential to avoid look-ahead bias. Survivorship bias must be addressed by including delisted firms. Data errors can drive spurious results.

8 Summary and Looking Ahead

This lecture has covered a lot of ground. Let us recap the key takeaways:

OLS fails with many predictors. When p is large relative to n, OLS has high variance and overfits. Even when $p < n$, the problem can be severe if signals are weak.

Penalized regression methods address this by shrinking coefficients. Ridge (L_2 penalty) shrinks all coefficients proportionally. Lasso (L_1 penalty) shrinks and can set coefficients exactly to zero, performing variable selection. Elastic Net combines both.

Shrinkage introduces bias but reduces variance. The net effect is often lower total MSE and better out-of-sample prediction.

Hyperparameter selection requires time-series cross-validation. Standard CV creates look-ahead bias with financial data. Always train on past, test on future.

Market timing is very hard. Even with regularization, we struggle to beat the historical mean. This is consistent with market efficiency.

Cross-sectional prediction is more promising. With more data and weaker efficiency assumptions, regularized methods can generate economically significant returns.

Statistical and economic performance can diverge. A model with better R^2_{OS} may have worse portfolio performance, and vice versa. Always evaluate economic value, not just statistical accuracy.

In the next lecture, we move beyond linear models to tree-based methods: decision trees, random forests, and gradient boosting. These methods can capture nonlinear relationships and interactions that linear models miss. A natural question: can nonlinear methods resurrect market timing, or is the failure fundamental?

Readings

Required:

- James, G., Witten, D., Hastie, T., Tibshirani, R., & Taylor, J. (2023). *An Introduction to Statistical Learning with Applications in Python*, Chapters 3, 6.2, and 6.4.

Supplementary:

- Goyal, A., Welch, I., & Zafirov, A. (2024). "A Comprehensive 2022 Look at the Empirical Performance of Equity Premium Prediction." *The Review of Financial Studies*. The definitive update on market timing with many predictors.
- Kozak, S., Nagel, S., & Santosh, S. (2020). "Shrinking the Cross-Section." *Journal of Financial Economics*. The case for Ridge regression in cross-sectional return prediction.
- Gu, S., Kelly, B., & Xiu, D. (2020). "Empirical Asset Pricing via Machine Learning." *The Review of Financial Studies*, Sections 1.2–1.3. Comprehensive comparison of ML methods for return prediction.